Analysis of Stacked and Regular Supervised Machine Learning Algorithms to Identify Corneal Astigmatism

Shreyans Jain[1], Mr. Lakhan Bhaskar Kadel[2]

[1]Class 10 Student, Crystal Springs Uplands Upper School; Hillsborough,
California 94010
[2]Research Scholar, Department of Computer Science; ICFAI University;
Jaipur, Rajasthan 302031
Email: [1]shjain27@crystal.org, [2]mr.lakhanbhaskar@gmail.com

Abstract: According to the National Library of Medicine **[2]**, astigmatism is an eye condition that affects more than 13 percent of the global population with refractive errors, or around 195 million people. Detecting astigmatism early, especially in children, can have a positive effect on their quality of life and vision strength. In this research, our goal is to create a machine learning (ML) model with multiple algorithms in a level-1 (*meta-model*) stacked **[3]** manner using logistic regression. Individual machine learning algorithms can sometimes be accurate; however, when multiple algorithms' capabilities are combined to create a stacked algorithm, the overall accuracy is increased to a value higher than that of any individual algorithm. First, because of there being so many different ML algorithms, we need to narrow down which ones to use in our stacked model. In this research, we use a decision tree classifier **[4]**, a random forest classifier **[5]**, the XGBoost algorithm **[6]**, and the Gradient Boosting classifier **[7]** in our stacked model. These algorithms were picked because of their individual accuracies being higher than other supervised ML algorithms tested, like the SVM **[16]**, the Naïve Bayes Classifier **[17]**, the ANN **[15]** (artificial neural network), and more. Finding the differences between these algorithms' scores can help us identify which algorithms are best suited to identify astigmatism and the reason behind them being successful in this specific task. Finding similarities between these successful algorithms can help us develop an even stronger stacked model. Overall, the central goal of this research is to develop a machine-learning model capable of accurately and efficiently identifying astigmatism in a medical scenario.

Keywords: Astigmatism, Machine Learning (ML), Meta Model, Regression, Logistic Regression, Decision Tree Classifier, Random Forest Classifier, XGBoost, Gradient Boosting, SVM, Naïve Bayes, and ANN.

1. Introduction

Astigmatism is a very prevalent eye refraction error in human populations as it affects more than 195 million people worldwide. According to the Cleveland Clinic **[9]**, Astigmatism can be hereditary and can be inherited from parents in fetal development. However, according to the National Library of Medicine **[8]**, astigmatism can also be developed through aging. According to the Mayo Clinic **[10]** and as seen in 'Figure 1', astigmatism is a treatable imperfection in the curvature of the eye that causes distorted blurry vision and nearsightedness.

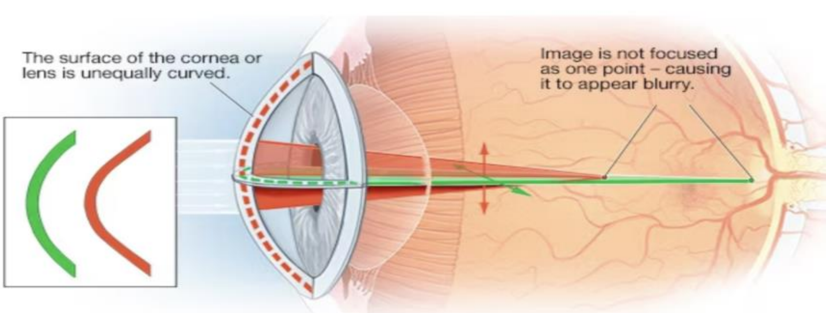

The surface of the cornea or lens is unequally curved.
Image is not focused as one point – causing it to appear blurry.

Figure 1 - A diagram of an astigmatism patient's eye from the Mayo Clinic

Astigmatism occurs when the cornea lens (which is normally a circle) becomes more oval-like. The earlier astigmatism is found and diagnosed, the less damage it can cause on a person's overall vision. Astigmatism can be corrected with the use of prescription spectacles.

Identifying astigmatism early is incredibly beneficial because it allows the imperfection to become correctable through LASIK **[11]** (a type of refractive laser eye surgery) and also decreases the risks for post-surgery issues and/or reactions. An accurate machine learning model that processes various traits to identify whether or not a patient has astigmatism can benefit not only many possessors but medical professionals as well who can “skip a step” in their process of diagnosis. Refractive eye diseases, especially astigmatism, can affect many peoples’ quality of life negatively, inhibiting them from doing many activities. The proper diagnosis of astigmatism for minors can allow for an early start of spectacles or contact lenses to stop the further development of the condition.

Various traits can allow for the identification of astigmatism. The dataset we used in our model allows for a smooth process of training and testing models due to the organization and preciseness of the contained data. In our model and in testing individual algorithms before creating the stacked one, an overall 80:20 data training to testing ratio was used. 80% of the data in the dataset went to training each algorithm/model while 20% of the data was used to test it. We tested with other values to find our training-to-testing ratio, but according to the papers we reviewed and our own findings, the 80:20 ratio remained the most optimal one.

Similar work in Opthalmology.

Our research was inspired by not only personal factors but other researchers who used different machine-learning techniques to identify astigmatism and other refractive issues in patients. The lack of similar research pushed us to continue researching this topic. Researchers we reviewed used similar machine learning algorithms in their models. First, Pablo Daniel Badillo et al. created a convolutional neural network (CNN) to detect astigmatism which analyzed photos of patients’ corneas after laser eye surgery. They were able to create a neural network to find patterns using their own dataset to detect astigmatism, obtaining an overall accuracy of 96.94 percent. The difference between their strategy and ours is that they take physical images whereas we rely on a model and verbal characteristics and descriptions to detect astigmatism.

In another paper reviewed, M. A. Valdes-Mas et al. **[13]**, used 2 ML algorithms to identify astigmatism in patients with keratoconus after an intracorneal ring implantation. Keratoconus, according to the Mayo Clinic **[14]** is an eye condition where the cornea gradually gets thinner and thinner, pointing out in a cone-like shape. Keratoconus creates blurry vision and can be corrected early with spectacles but requires a cornea transplant later on to treat. M. A. Valdes-Mas et al. created an artificial neural network **[15]** (ANN) and a decision tree classifier **[4]** (similar to the random forest classifier). They used two relatively medium-sized datasets to obtain an overall accuracy of 92% after ruling out the first dataset as it did not allow for accurate results.

Another group of researchers, Athene Forschung et al. **[18]**, created a deep learning model to detect astigmatism with microfluidic devices and overlapping particle imagery; with random sound and no sound. They obtained an overall accuracy of 96%. Although their method is much different from our method of detecting, the main factor being that their method relies on advanced imagery, connections can still be made with both. First, both logistic regression classifiers (that we are using in this research) and deep learning models **[19]** share similarities of being non-linear models. They are also made up of interconnected nodes in layers.

Used Dataset.

The dataset used in this research, as shown in ‘Table 1’, was obtained from the UCI machine learning data repository **[1]**. The dataset contains 5 variables, one being a binary variable holding the data of whether or not a patient with specific traits in other variables is astigmatic (1) or not (0). The other variables (excluding the ID) allow for our model to identify patients with astigmatism. The original “target” in this dataset is the class. The class was supposed to identify whether or not the patient should wear contact lenses and if yes, what type. The age variable contains information about whether the patient is young, pre-presbyopic, or presbyopic: a patient that is presbyopic is, according to the Mayo Clinic **[20]**, at the natural age for the eye to deteriorate. This dataset contains information from 24 patients. ML algorithms are able to analyze and find patterns in this dataset. An individual algorithm would not be able to come up with a highly accurate result based on data from 24 people. However, a stacked model consisting of results from various algorithms is able to generate accurate results.

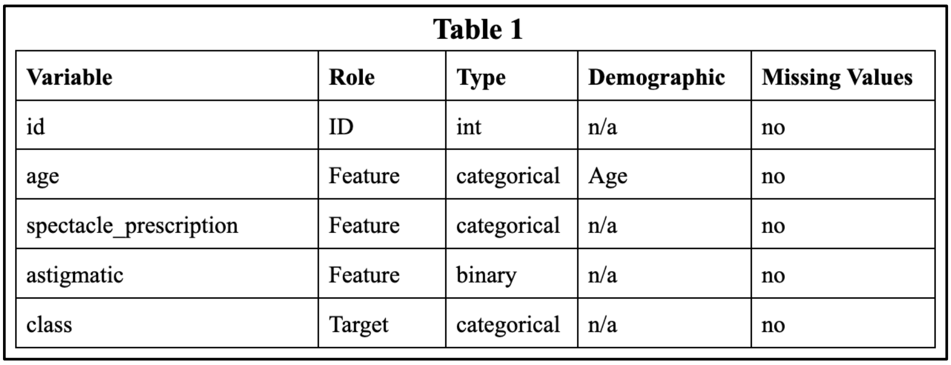


Table 1

Variable	Role	Type	Demographic	Missing Values
id	ID	int	n/a	no
age	Feature	categorical	Age	no
spectacle_prescription	Feature	categorical	n/a	no
astigmatic	Feature	binary	n/a	no
class	Target	categorical	n/a	no

Our stacked model.

The stacked model **[3][21]** we produced (a logistic regression classifier **[22]**) was trained on the results of a decision tree classifier **[4]**, a random forest classifier **[5]**, the XGBoost algorithm **[6]**, and a Gradient Boosting classifier **[7]**. The model was run using Python 3.12 on a Macbook M2 Pro in Visual Studio Code. First, each individual algorithm was trained on the data, each producing different results. The data was divided (shuffled) into 5 different subsets (5 validation folds **[23]**) and the base individual models were each trained on it, 5 times in total. These features, known as meta-features **[24]** are what the stacked model (the logistic regression classifier), trains itself on. This stacking allows for the grouped strengths of each base model to be combined while the individual weaknesses are mitigated by the generation of the meta-data. These steps can be seen in ‘Figure 2’.

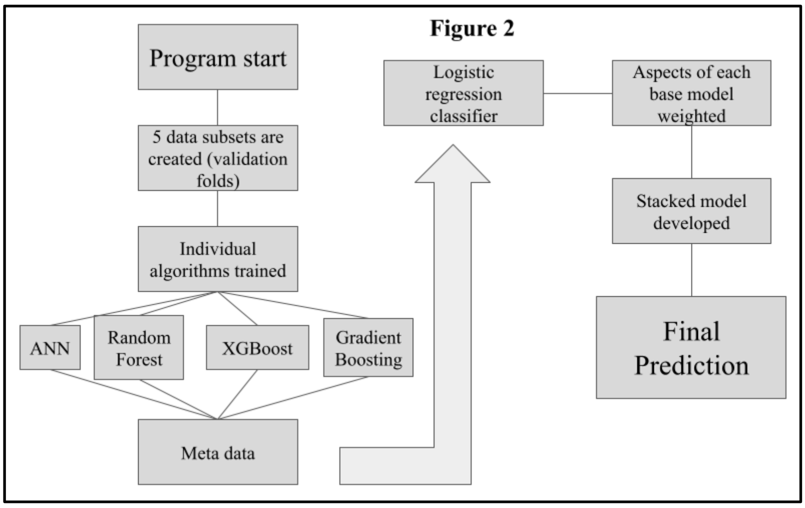


Figure 2
Program start
5 data subsets are created (validation folds)
Individual algorithms trained
ANN
Random Forest
XGBoost
Gradient Boosting
Meta data
Logistic regression classifier
Aspects of each base model weighted
Stacked model developed
Final Prediction

Logistic Regression.

Overall, logistic regression **[22]** uses a linear approach to calculate a binary value (in our case astigmatism - 1 for yes and 0 for no). The classifier assumes a linear relationship between multiple variables and then creates a “line” between the two or more sub-groups that it identifies that link to the target value. Then, to convert this line into an accurate prediction method, the classifier uses the Sigmoid function (or logistic function) which looks like an S-shape curve. It then is able to separate inputted information into groups: in our case 1 for if the patient has astigmatism and 0 for if the patient’s eyes are normal.

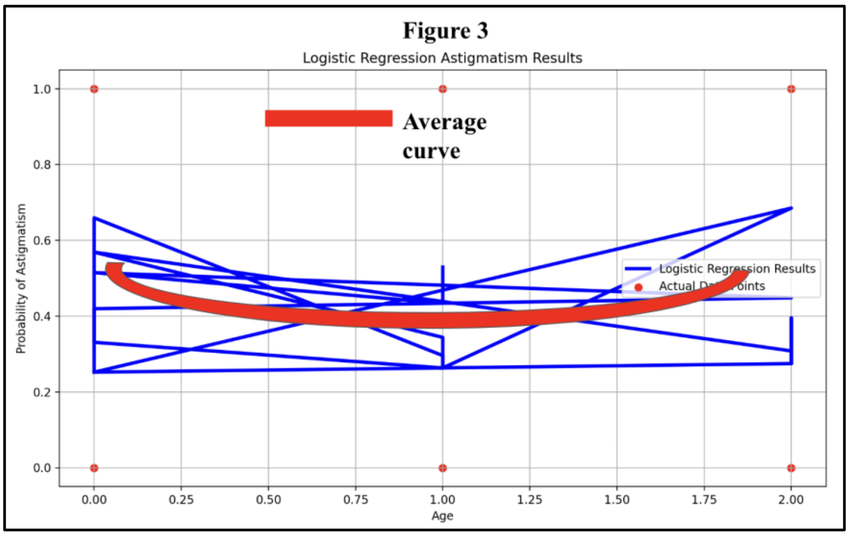


Figure 3

Random Forest Classifier.

The random forest classifier **[5]** consists of many binary trees (hence its name: forest). A binary tree consists of a central “branch” that asks a yes/no question: 1/0. If the question is answered with a 0 (in this case, let’s use an example of if the person is young or not), the next branch that will be accessed will ask a different question than

the branch traveled to if the question was answered with a 1. Eventually, all data is considered and a final conclusion is reached: 1 or 0: astigmatism or not. The process of a binary tree can be seen in ‘Figure 4’.
However, one simple binary tree is extremely inaccurate when compared with large amounts of data. So, for every new ID/instance in the dataset, a new tree is created/trained. The random forest classifier takes all of the results of the trees and averages them, as seen in ‘Figure 5’ (each box represents an individual binary tree with 1 meaning that the patient has astigmatism and 0 meaning that the patient does not). This average is far more accurate than just a singular binary tree as it represents the power of many binary trees at once.

Figure 4

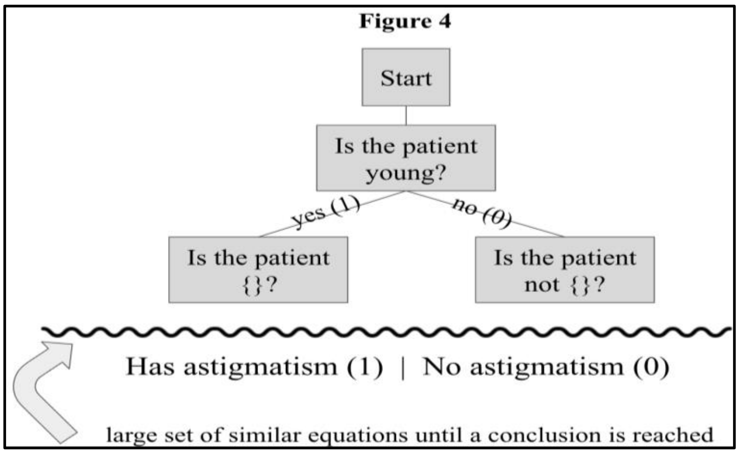
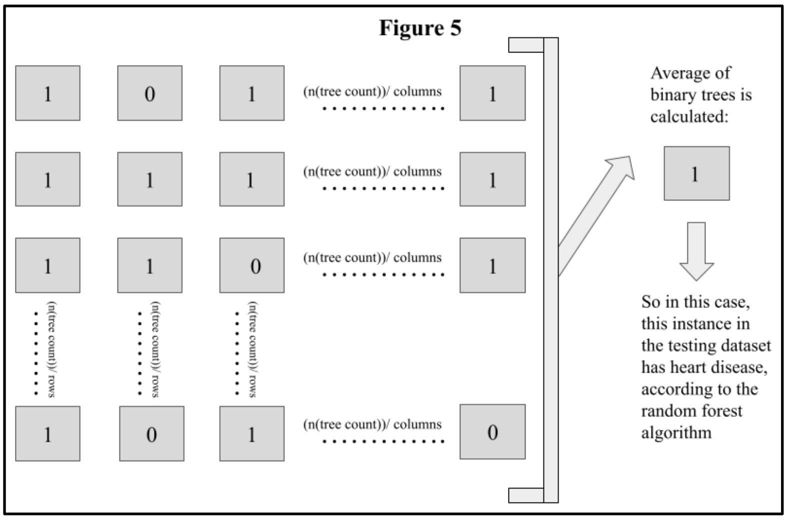

Start
Is the patient young?
yes (1)
no (0)
Is the patient {}?
Is the patient not {}?
Has astigmatism (1) | No astigmatism (0)
large set of similar equations until a conclusion is reached

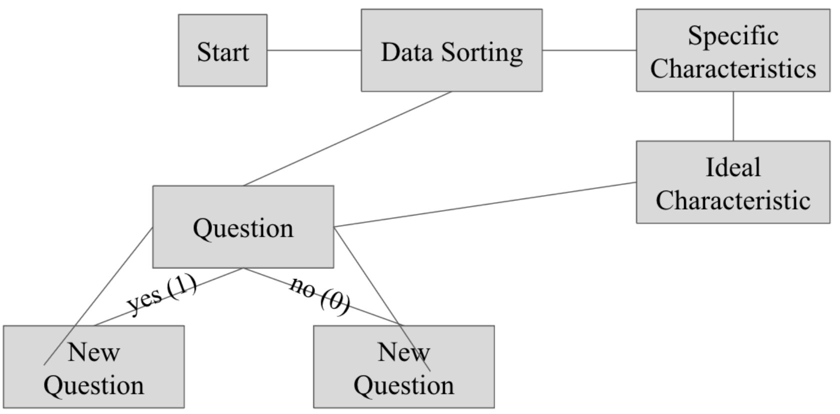
Figure 5



Decision Tree Classifier.

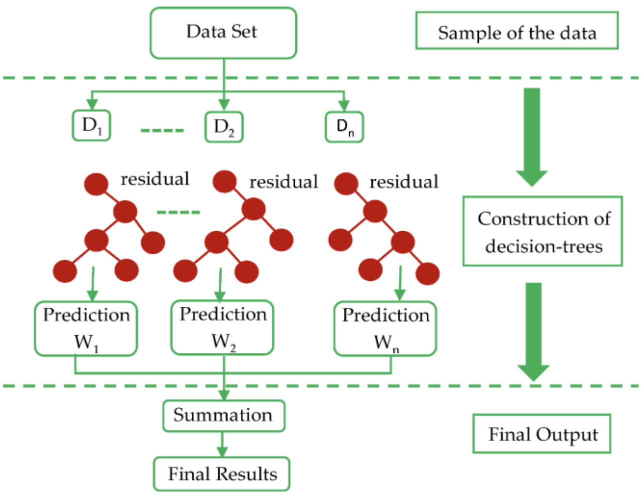
With a very similar name to the random forest classifier, the decision tree classifier **[4]** works in a somewhat similar way. However, there are some key differences. First, the classifier works to split the data so that it can work with it efficiently. The algorithm splits the training data based on how it is organized and its level of order. Overall, the algorithm selects a feature that best divides the data (in this case astigmatism). Then, a binary tree is created which works in the same way previously stated: “yes or no” questions are asked to narrow down an answer. However, the key difference is the resorting of the training data based on the answer to the question each time, eliminating other options completely, as seen in ‘Figure 6’.

Figure 6

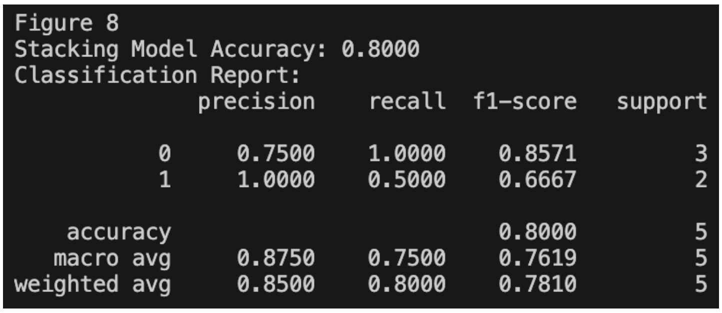


Extreme Gradient Boosting Algorithm.
An extreme gradient boosting algorithm **[6]** or "XGBoost" works by using decision trees. In our case, the algorithm starts with various features and a target variable: astigmatism or not. The algorithm first sorts the data so that its performance is more efficient. It then linearly predicts the mean value of whether or not a person has astigmatism using the training data. Then, the algorithm creates a decision tree which runs normally on an instance of the data, later returning back to the original guess. It then corrects the mistakes in the average; this is only the starting steps. Based on the mistake(s), it splits the data at (a) very specific point(s) greedily: with an accurate splitting tool like the greedy approximation algorithm. In our case, the algorithm keeps going until the amount of instances have all been used up. The resulting model (or set of binary trees) have been changed enough for them to be extremely accurate as a group, as seen in 'Figure 7 **[25]**'.

Figure 7



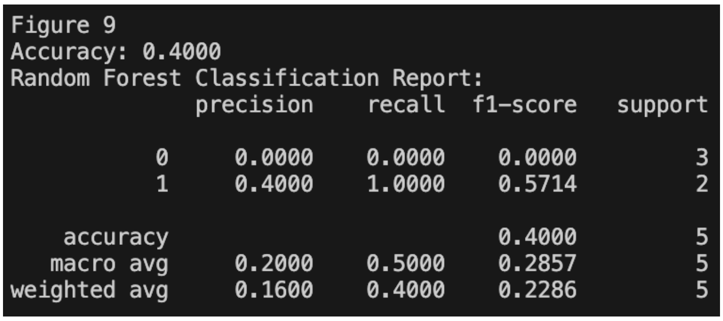
Gradient Boosting Algorithm.
The gradient boosting algorithm **[7]** works in the same way as the extreme gradient boosting algorithm (or XGBoost). It builds upon its own work, correcting its mistakes as it progresses through the training data. The main difference between the two algorithms is that the extreme gradient boosting algorithm uses specific algorithms to split nodes and also has built-in handling of missing values.



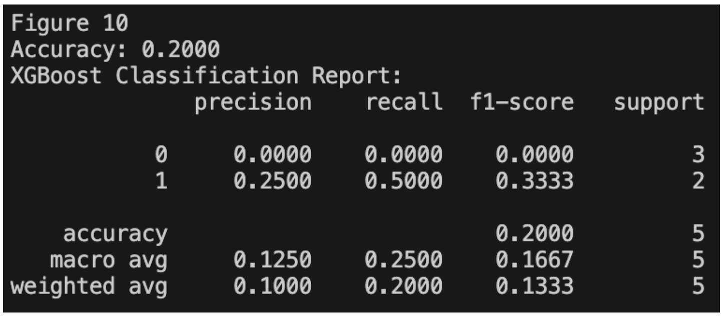
```
Figure 8
Stacking Model Accuracy: 0.8000
Classification Report:
              precision    recall  f1-score   support

           0     0.7500    1.0000    0.8571         3
           1     1.0000    0.5000    0.6667         2

    accuracy                         0.8000         5
   macro avg     0.8750    0.7500    0.7619         5
weighted avg     0.8500    0.8000    0.7810         5
```

2. Results & Conclusion.

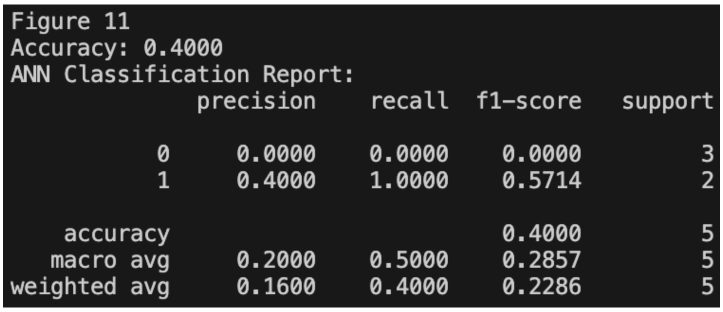
As stated previously, after running all of the individual algorithms, a stacked model was created using linear regression that had an accuracy rating of more than double than that of any individual algorithm. The used dataset **[1]** was split into training and testing subsets with an 80:20 size ratio. The individual algorithms researched in this paper were the decision tree classifier **[4]**, a random forest classifier **[5]**, the XGBoost algorithm **[6]**, and a Gradient Boosting classifier **[7]**. The algorithms narrowed down after testing various other supervised ones all shared one trait: they used binary (regression) trees. This specific shared success in used algorithms is because machine learning algorithms with binary trees are able to partition and split data extremely efficiently, especially when the amount of individual variables is low **[26]**. The overall accuracy of the logistic regression classifier was 80.0% as seen in 'Figure 8'. For each of the individual algorithms, the random forest classifier scored 40.0% as seen in 'Figure 9'. The XGBoost algorithm obtained an accuracy score of just 20.0%, as seen in 'Figure 10'. The regular gradient boosting algorithm scored a surprising 40.0% as seen in 'Figure 11'. Finally, the decision tree classifier scored a 40.0% overall accuracy rating, as seen in 'Figure 12'. Overall, a stacked model with multiple different algorithms was able to produce an accuracy rating higher than most individual algorithms because of its adaptation of the strengths of individual algorithms. Similarly, in our case, the stacked algorithm we used was like another step in the entire process of using binary trees: since all of our algorithms had a central goal of using binary trees to remove errors in their predictions, the stacked model just did that again: it used the data from 4 algorithms that had stacks of binary trees to create a singular model, just like the individual algorithms. It erased mistakes by cross-checking each of the individual algorithms' results, generating an overall accuracy rating of more than double of any individual algorithms' accuracy, as seen in 'Figure 13'.



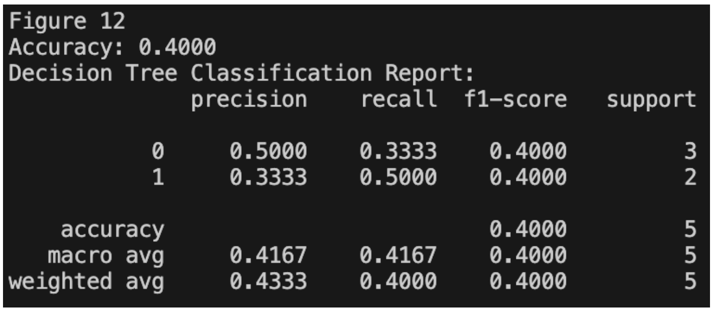
```
Figure 9
Accuracy: 0.4000
Random Forest Classification Report:
              precision    recall  f1-score   support

           0     0.0000    0.0000    0.0000         3
           1     0.4000    1.0000    0.5714         2

    accuracy                         0.4000         5
   macro avg     0.2000    0.5000    0.2857         5
weighted avg     0.1600    0.4000    0.2286         5
```



```
Figure 10
Accuracy: 0.2000
XGBoost Classification Report:
              precision    recall  f1-score   support

           0     0.0000    0.0000    0.0000         3
           1     0.2500    0.5000    0.3333         2

    accuracy                         0.2000         5
   macro avg     0.1250    0.2500    0.1667         5
weighted avg     0.1000    0.2000    0.1333         5
```



```
Figure 11
Accuracy: 0.4000
ANN Classification Report:
              precision    recall  f1-score   support

           0     0.0000    0.0000    0.0000         3
           1     0.4000    1.0000    0.5714         2

    accuracy                         0.4000         5
   macro avg     0.2000    0.5000    0.2857         5
weighted avg     0.1600    0.4000    0.2286         5
```



```
Figure 12
Accuracy: 0.4000
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0     0.5000    0.3333    0.4000         3
           1     0.3333    0.5000    0.4000         2

    accuracy                         0.4000         5
   macro avg     0.4167    0.4167    0.4000         5
weighted avg     0.4333    0.4000    0.4000         5
```

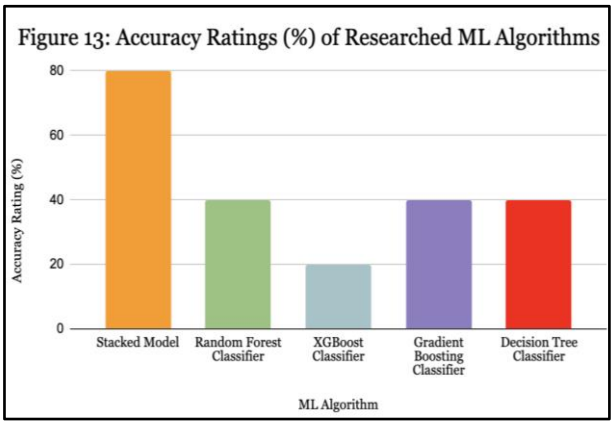


Figure 13: Accuracy Ratings (%) of Researched ML Algorithms

3. References & Bibliography

1. Cendrowska, J. "Lenses." UC Irvine, 31 July 1990. *UCI Machine Learning Repository*, https://doi.org/10.24432/C5K88Z. Accessed 17 July 2024. Data set.
   This is the dataset used in our research.
2. Gurnani, Bharat, and Kirandeep Kaur. *Astigmatism*. *National Library of Medicine; National Center for Biotechnology Information*, StatPearls Publishing LLC, 26 June 2023, www.ncbi.nlm.nih.gov/books/NBK582142/. Accessed 17 July 2024.
3. Brownlee, Dr. Jason. "Stacking Ensemble Machine Learning With Python."*Machine Learning Mastery*, 27 Apr. 2021, machinelearningmastery.com/stacking-ensemble-machine-learning-with-python/. Accessed 17 July 2024.
4. Brownlee, Dr. Jason. "k-Fold Cross Validation" *Machine Learning Mastery*, 4 Oct. 2023, www.machinelearningmastery.com/k-fold-cross-validation/. Accessed 17 July 2024.
5. IBM, editor. "What is a Decision Tree?" *IBM*, International Business Machines, 18 Apr.
6. 2022, www.ibm.com/topics/decision-trees. Accessed 12 July 2024.
7. —, editor. "What is random forest?" *IBM*, International Business Machines, 7 Mar. 2023, www.ibm.com/topics/random-forest. Accessed 12 July 2024.
8. —, editor. "What is XGBoost?" *IBM*, International Business Machines, Mar. 2023, www.ibm.com/topics/xgboost. Accessed 12 July 2024.
9. —, editor. "What is boosting?" *IBM*, International Business Machines, 17 Jan. 2023, www.ibm.com/topics/boosting. Accessed 12 July 2024.

10. —, editor. "What are neural networks?" *IBM*, International Business Machines, 16 Dec. 2021, hwww.ibm.com/topics/neural-networks. Accessed 12 July 2024.
11. —, editor. "How SVM Works." *IBM*, International Business Machines, 20 Dec. 2022, www.ibm.com/docs/en/spss-modeler/saas?topic=models-how-svm-works. Accessed 12 July 2024.
12. —, editor. "What are Naïve Bayes Classifiers?" *IBM*, International Business Machines, 18 Mar. 2023, www.ibm.com/topics/naive-bayes. Accessed 12 July 2024.
13. —, editor. "What is deep learning?" *IBM*, International Business Machines, 25 Dec. 2022, www.ibm.com/topics/deep-learning. Accessed 1 July 2024.
14. —, editor. "What is logistic regression?" *IBM*, International Business Machines, 16 Oct. 2021, www.ibm.com/topics/logistic-regression. Accessed 14 July 2024.
15. Namba, Hiroyuki, et al. "Age-Related Changes in Astigmatism and PotentialCauses." *Cornea*, vol. 39, no. 1, 7 Oct. 2020, pp. S34-S38, https://doi.org/10.1097/ico.0000000000002507.
16. Cleveland Clinic, editor. "Astigmatism." *The Cleveland Clinic*, 20 Nov. 2011,my.clevelandclinic.org/health/diseases/8576-astigmatism. Accessed 17 July2024.
17. Mayo Clinic Staff. "Astigmatism." *The Mayo Clinic*, 5 Oct. 2021,www.mayoclinic.org/diseases-conditions/astigmatism/symptoms-causes/syc-20353835.Accessed 12 June 2024.
18. —. "Astigmatism." *The Mayo Clinic*, 5 Oct. 2021,www.mayoclinic.org/diseases-conditions/keratoconus/symptoms-causes/syc-20351352.Accessed 17 July 2024.
19. —. "Presbyopia - Symptoms and Causes." *Mayoclinic*, Mayo Foundation for MedicalEducation and Research, 20 Nov. 2021, www.mayoclinic.org/diseases-conditions/presbyopia/symptoms-causes/syc-20363328. Accessed 17July 2024.
20. Boyd, Kierstan. "LASIK - Laser Eye Surgery." Edited by David Turbert. *TheAmerican Academy of Opthalmology*, 25 May 2024, www.aao.org/eye-health/treatments/lasik. Accessed 1 July 2024.
21. Badillo, Pablo Daniel, et al. "Convolutional Neural Networksfor Astigmatism Detection." IEEE Conference of Russian Young Researchers inElectrical and Electronic Engineering (EIConRus), 2022.
22. Valdes-Mas, M. A., et al. "Machine learning for predicting astigmatism inpatients with keratoconus after intracorneal ring implantation." IEEE-EMBSInternational Conference on Biomedical and Health Informatics (BHI), 2014.*ResearchGate*, June 2014, www.researchgate.net/publication/269291078_Machine_learning_for_predicting_astigmatism_in_patients_with_keratoconus_after_intracorneal_ring_implantation. Accessed 18 July 2024.
23. Forschung, Athene, et al. "On the use of machine learning algorithms for the calibration of astigmatism PTV." Universität der Bundeswehr München, 2020. www.athene-forschung.unibw.de/doc/129121/129121.pdf. Accessed 19 July 2024
24. Javatpoint. "Stacking in Machine Learning." *Javatpoint*, www.javatpoint.com/stacking-in-machine-learning. Accessed 17 July 2024.
25. Rivolli, Adriano, et al. "Meta-features for Meta-learning." *Knowledge-BasedSystems*, vol. 240, Mar. 2022, p. 108101, https://doi.org/10.1016/j.knosys.2021.108101. Accessed 17 July 2024.
26. Amjad, Maaz, et al. "Prediction of Pile Bearing Capacity Using XGBoostAlgorithm: Modeling and Performance Evaluation." *Applied Sciences*, vol. 12,no. 4, 18 Feb. 2022, p. 2126, https://doi.org/10.3390/app12042126.
27. Geeks for Geeks, editor. "Applications, Advantages and Disadvantages of BinaryTree." *GeeksforGeeks*, 29 Mar. 2023, www.geeksforgeeks.org/applications-advantages-and-disadvantages-of-binary-tree/. Accessed 17 July2024.